DOCUMENT RESUME

ED 183 402 SE 030 311

AUTHOR Brotherton, Sheila: And Others
TITLE Boolean Algebra. Geometry Module for Use in a Mathematics Laboratory Setting.
INSTITUTION Regional Center for Pre-Coll. Mathematics, Denver, Colo.
SPONS AGENCY National Science Foundation, Washington, D.C.
PUB DATE 74
GRANT NSF-GW-7720
NOTE 68p.: For related documents, see SE 030 304-322: Contains light and broken type: Computer print-outs marginally legible

EDRS PRICE MF01/PC03 Plus Postage.
DESCRIPTORS *Activities: Computer Oriented Programs: Electric Circuits: Geometric Concepts: Geometry: *Learning Laboratories: Logic: Mathematical Applications: *Mathematical Enrichment: *Mathematical Logic: Mathematics Curriculum: *Mathematics Instruction: Secondary Education: *Secondary School Mathematics: Worksheets

ABSTRACT

This module is recommended as an honors unit to follow a unit on logic. There are four basic parts: (1) What is a Boolean Algebra: (2) Using Boolean Algebra to Prove Theorems: (3) Using Boolean Algebra to Simplify Logical Statements: and (4) Circuit Problems with Logic and Boolean Algebra. Of these, sections 1, 2, and 3 are primarily written exercises. Section 4 involves modeling problems on a circuit board. Some supplementary materials are included at the end of the module. An additional section on computer-extended Boolean Algebra appears at the end of the teacher's guide. (Author/MK)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Mary L. Charles
NSF
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

ED183402

GEOMETRY MODULE FOR USE IN A MATHEMATICS LABORATORY SETTING

BOOLEAN ALGEBRA

by

Sheila Brotherton
Glenn Bruckhart
James Reed

A Publication of

University of Denver
Mathematics Laboratory
Regional Center for
Pre-College Mathematics

Dr. Ruth I. Hoffman, Director

This material was prepared with the support of the National Science Foundation Grant #GW-7720.

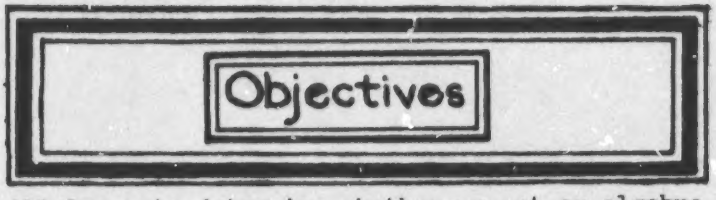
c University of Denver Mathematics Laboratory

SE 030 311

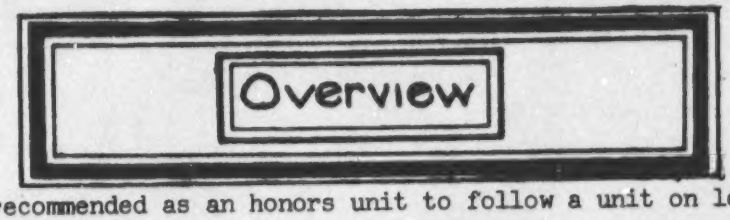
00133

BOOLEAN ALGEBRA

Teacher Guide



1. The student will learn to determine whether or not an algebra is a Boolean Algebra.
2. The student will be able to use the axioms of Boolean Algebra to prove theorems, simplify logical statements and simplify circuits.



This module is recommended as an honors unit to follow a unit on logic. There are four basic parts:

I. What is a Boolean Algebra?

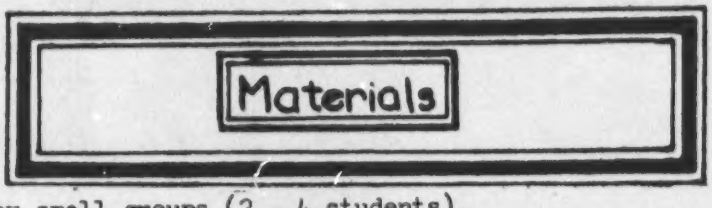
II. Using Boolean Algebra to prove Theorems.

III. Using Boolean Algebra to Simplify Logical Statements.

IV. Circuit Problems with Logic and Boolean Algebra

Of these sections I, II, III, are primarily written exercises. Section IV involves modeling problems on a circuit board.

Some supplementary problems are included at the end of the module.



Circuit boards for small groups (2 - 4 students).

An additional section on computer extended Boolean Algebra appears at the end of the teacher's guide.

© '74 U of DML

I. What is a Boolean Algebra?

Teacher Guide

Teaching Suggestions

1. Stronger students will be able to take the information card and proceed on their own to the exercises, or the teacher may wish to use the information card as a class discussion.
2. The exercises can be done individually or in small groups.

© '74 U of DML

What is a Boolean Algebra?

Exercise Answers

Boolean Algebra Laws for Logic $(P, \vee, \wedge)$

1.	Closure Laws	$p \vee q \in P$	$p \wedge q \in P$
2.	Commutative Laws	$p \vee q \leftrightarrow q \vee p$	$p \wedge q \leftrightarrow q \wedge p$
3.	Associative Laws	$p \vee (q \vee r) \leftrightarrow (p \vee q) \vee r$	$p \wedge (q \wedge r) \leftrightarrow (p \wedge q) \wedge r$
4.	Distributive Laws	$p \vee (q \wedge r) \leftrightarrow (p \vee q) \wedge (p \vee r)$	$p \wedge (q \vee r) \leftrightarrow (p \wedge q) \vee (p \wedge r)$
5.	Identity Laws	there exists identity $f \in P$ such that $p \vee f \leftrightarrow p$	there exists identity $t \in P$ such that $p \wedge t \leftrightarrow p$
6.	Complementation Law	for every $p \in P$ there exists $\sim p \in P$ such that $p \vee \sim p \leftrightarrow t$	for every $p \in P$ there exists $\sim p \in P$ such that $p \wedge \sim p \leftrightarrow f$

© '74 U of DML

Exercise Answers

1. $(C,+,*)$ is a Boolean Algebra

Identity

0 is + identity	1 is * identity
$1 + 0 = 1$	$1 * 1 = 1$
$0 + 0 = 0$	$0 * 1 = 0$

Complementation

$-1 = 0$	$1^{-1} = 0$
$-0 = 1$	$0^{-1} = 0$
$1 + -1 = 1$	$1 * 1^{-1} = 0$
$0 + -0 = 1$	$0 * 0^{-1} = 0$

2. $(S,\cup,\cap)$ is a Boolean Algebra

Identity

Let A be a set	Let A be a set
$\emptyset$ the empty set	U be universal set
$A \cup \emptyset = A$	$A \cap U = A$

Complementation

Let $\overline{A}$ be complement of A:	
$A \cup \overline{A} = U$	$A \cap \overline{A} = \emptyset$

3. $(R,+,x)$ is not a Boolean Algebra

Identity

Let $r \in R$, 0 the add. identity, 1 the multiplication identity.

$r + 0 = r$	$r \times 1 = r$

Complementation

Let $-r$ be add. inverse of r	Let $1/r$ be multiplication inverse of r
$-r + r = 0$	$1/r \times r = 1$

Complementation Law doesn't hold.

© '74 U of DML

II. Using Boolean Algebra to Prove Theorems

Teacher Guide

Teaching Suggestions

1. Some students will be able to read this information card and then write their own proofs. Most however, will need teacher help to understand the example proof.
2. Students can work on the proofs individually or in groups of 2 or 3.

© '74 U of DML

Exercise Answers

Theorem 1

1. $p \vee f \leftrightarrow p$	1. Identity $\vee$
2. $p \vee (p \wedge \sim p) \leftrightarrow p$	2. Complementation $\wedge$
3. $(p \vee p) \wedge (p \vee \sim p) \leftrightarrow p$	3. Distributive $\vee$ over $\wedge$
4. $(p \vee p) \wedge t \leftrightarrow p$	4. Complementation $\vee$
5. $p \vee p \leftrightarrow p$	5. Identity $\wedge$

Theorem 2

1. $p \wedge t \leftrightarrow p$	1. Identity $\wedge$
2. $p \wedge (p \vee \sim p) \leftrightarrow p$	2. Complementation $\vee$
3. $(p \wedge p) \vee (p \wedge \sim p) \leftrightarrow p$	3. Distributive $\wedge$ over $\vee$
4. $(p \wedge p) \vee f \leftrightarrow p$	4. Complementation $\wedge$
5. $(p \wedge p) \leftrightarrow p$	5. Identity $\vee$

Theorem 3

1. $p \vee \sim p \leftrightarrow t$	1. Complementation $\vee$
2. $(p \vee p) \vee \sim p \leftrightarrow t$	2. Theorem 1
3. $p \vee (p \vee \sim p) \leftrightarrow t$	3. Associative $\vee$
4. $p \vee t \leftrightarrow t$	4. Complementation $\vee$

Theorem 4

1. $p \wedge \sim p \leftrightarrow f$	1. Complementation $\wedge$
2. $(p \wedge p) \wedge \sim p \leftrightarrow f$	2. Theorem 2
3. $p \wedge (p \wedge \sim p) \leftrightarrow f$	3. Associative $\wedge$
4. $p \wedge f \leftrightarrow f$	4. Complementation $\wedge$

Theorem 5

1. $p \wedge (p \wedge q) \leftrightarrow (p \wedge t) \vee (p \wedge q)$	1. Identity $\wedge$
2. $\leftrightarrow p \wedge (t \vee q)$	2. Distributive $\wedge$ over $\vee$
3. $\leftrightarrow p \wedge (t)$	3. Theorem 3
4. $\leftrightarrow p$	4. Identity $\wedge$

Theorem 6

1. $p \wedge (p \vee q) \leftrightarrow (p \vee f) \wedge (p \vee q)$	1. Identity $\vee$
2. $\leftrightarrow p \vee (f \wedge q)$	2. Distributive $\vee$ over $\wedge$
3. $\leftrightarrow p \vee (f)$	3. Theorem 4
4. $\leftrightarrow p$	4. Identity $\vee$

© '74 U of DML

III. Using Boolean Algebra to Simplify Logical Statements
Teacher Guide

Teaching suggestions

1. Some can proceed without help, but most will need a teacher explanation of the example.
2. Point out that statements proven in the last unit can now be used as reasons.
3. Students can work individually or in small groups of two or three.

© '74 U of DML

Simplifying with Boolean Algebra

Exercise Answers

1. $p \vee (p \wedge q)$

   $(p \wedge t) \vee (p \wedge q)$

   $p \wedge (t \vee q)$

   $p \wedge t$

   p

2. $(p \wedge q) \vee [(p \vee \sim p) \wedge \sim q]$

   $(p \wedge q) \vee [t \wedge \sim q]$

   $(p \wedge q) \vee [\sim q]$

   $(p \vee \sim q) \wedge (q \vee \sim q)$

   $(p \vee \sim q) \wedge t$

   $p \vee \sim q$

3. p: My teacher is boring

   q: My teacher is funny

   $(p \vee q) \wedge (\sim p \wedge \sim q)$

   $[p \wedge (\sim p \wedge \sim q)] \vee [q \wedge (\sim p \wedge \sim q)]$

   $[(p \wedge \sim p) \wedge \sim q] \vee [(q \wedge \sim q) \wedge \sim p]$

   $(f \wedge \sim q) \vee (f \wedge \sim p)$

   f $\vee$ f

   f

© '74 U of DML

Circuit Problems with Logic and Boolean Algebra

Teacher Guide

Teaching Suggestions

1. Small groups of 3 or 4 can work together to model the problems in this activity.
2. Problems following the activity can be done by individuals or groups.

Materials

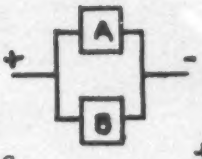
Logic Circuit Boards for each group.

© '74 U of DML

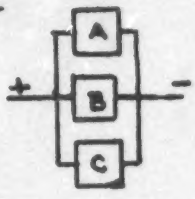
Circuit Problems with Logic and Boolean Algebra

Exercise Answers

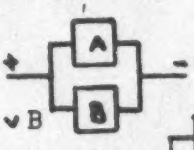
1. $A \vee B$



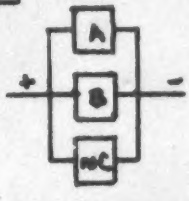
2. $A \vee B \vee C$



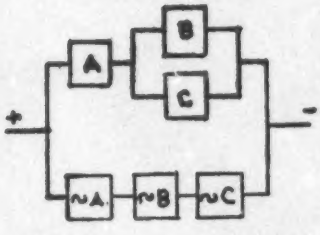
3. $A \vee B$



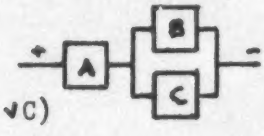
4. $A \vee B \vee \sim C$



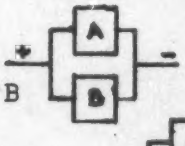
5. $[A \wedge (B \vee C)] \vee [\sim A \wedge \sim B \wedge \sim C]$



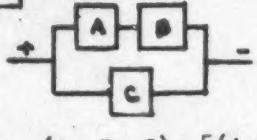
6. $A \wedge (B \vee C)$



7. $A \vee B$



8. $(A \wedge B) \vee C$



9. Circuit I: $(A \wedge B \wedge C) \vee [(A \vee B \wedge C) \wedge D]$

Circuit II: $(A \wedge D) \vee (A \wedge B \wedge C) \vee (D \wedge B \wedge D) \vee (D \wedge C)$
$(A \wedge B \wedge C) \vee (A \wedge D) \vee (B \wedge D) \vee (C \wedge D)$
$(A \wedge B \wedge C) \vee [(A \wedge B \wedge C) \wedge D]$

10. $(D \vee B) \wedge [(C \wedge D) \vee (\sim C \wedge \sim D)]$
11. $D \vee (B \wedge \sim C)$
12. $A \vee C \vee (\sim D \wedge \sim B)$
13. $(A \vee C) \wedge (B \vee D)$

© '74 U of DML

Supplementary Exercises

Answers

1. Suppose $A \cap C \neq \emptyset$, then $\exists$ an element say $x \in A \cap C$. By the definition of $\cap$ $x \in A$ and $x \in C$. Now since $A \subset B$, $x \in B$. Therefore $B \cap C = x$. But this contradicts the fact that $B \cap C = \emptyset$. Therefore $A \cap C \neq \emptyset$ must be false and $A \cap C = \emptyset$, true.

2. a) Suppose x is even, then x^2 is even. But x^2 is odd. Therefore "x is even" must be false and x is odd, true.

   b) Suppose I do not do homework regularly, then I will not pass the course. Curses, I must pass the course to graduate. Therefore "I do not do homework regularly" is false and I must do homework regularly, true.

   c) Suppose he is a college professor, then he does not earn a great deal of money (more than \$30,000). But this contradicts the fact that he earns a great deal of money. Therefore the supposition must be false and "he is not a college professor", true.

3. Let s represent the statement "he is to succeed."
   " c " " " "he must be competent."
   " l " " " "he must be lucky."
   " w " " " "something is sure to go wrong."

   $s \rightarrow (c \wedge l)$

   $\sim c \rightarrow \sim s$

   $\sim l \rightarrow w$

4. Let m represent the statement "Jones is the murderer."
   " t " " " "he knows the exact time of death."
   " w " " " "he knows the murder weapon."

   $m \rightarrow (t \wedge w)$

   $(\sim t \vee \sim w) \rightarrow \sim m$.

© '74 U of DML

5.

p	q	$p \rightarrow q$	$p \wedge \sim q$	$(p \wedge \sim q) \rightarrow \sim p$	$(p \wedge \sim q) \rightarrow q$	$(p \wedge \sim q) \rightarrow (\sim\sim q)$
T	T	T	F	T	T	T
T	F	F	T	F	F	F
F	T	T	F	T	T	T
F	F	T	F	T	T	T
		(1)		(a)	(b)	

6. g → b

lf → t

op → ∿sc

b → if

wkg → g

∿sc → ∿t

Conclusion: If he wears white kid gloves, then he is not an opium eater.

7. Does the first door lead to freedom if and only if you are truthful?

© '74 U of DML

Computer Extended Boolean Algebra

The Computing and Mathematics Curriculum Project sponsored by the Department of Mathematics, University of Denver has developed a series of modules that extend mathematics via the computer.

Application of Logic to Circuit Design by John Clark, Irwin Hoffman, and William Talbert offers an excellent opportunity for students to interact with a computer to utilize Boolean Algebra in designing circuits.

That module is included here along with the original authors' teacher suggestions, procedures, material, and student assignments.

If the Boolean Algebra unit is to follow the logic unit early in the school year, this computer section will more than likely be much too sophisticated for the majority of geometry students. Sometime during the second semester this material could readily be used.

© '74 U of DML

COMPUTING AND MATHEMATICS CURRICULUM PROJECT

CMCP

Computer Extended Mathematics

6 - APPLICATION OF LOGIC TO CIRCUIT DESIGN

A Student Assignment Unit

For High School Enrichment

With Suggestions for the Teacher

DIRECTED BY: Dr. William S. Dorn
Dr. Ruth I. Hoffman

August, 1971

© '74 U of DML

6 - APPLICATION OF LOGIC TO CIRCUIT DESIGN

Use of ASSIGNMENT Units

This is a single topic unit which is part of a series of selected subjects that may be matched to any textbook and used in any order. A teacher may wish to make a computer assignment to accompany his class presentation for preview, review or supplementary work.

NOTE: We would like to express our appreciation to Alan and Mark Frank, students at George Washington High School, Denver, Colorado, for providing the general ideas developed in this unit, as well as contributing to the student assignment.

© '74 U of DML

TEACHER SUGGESTIONS

Assumptions

1. The student is either familiar with truth tables and the use of logic to prove tautologies or is at a stage in his mathematical development that teaching him this material would be useful.
2. The teacher is willing to let the student manipulate polynomials in Boolean Algebra so that the concept of $a \cdot a = a$ (the idempotent law), where a is an element of the Boolean Algebra is meaningful.

Example:

$$\begin{aligned} a^3 &= a(a \cdot a) \\ &= a(a) \\ &= a \cdot a \\ &= a \end{aligned}$$

Computer Assignment

The purpose of this unit is to show the student an application of logic and truth tables to the design of circuits. In this endeavor we develop the concept of AND gates, EXCLUSIVE OR gates, and OR gates. We then represent the entire circuit as a circuit polynomial which may need to be reduced by manipulations in Boolean Algebra.

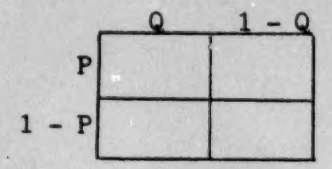
Procedures

1. Mimeograph the material developed here as it is something that generally will not appear in a high school text book. Students might find it interesting to have available.

© '74 U of DML

2. Development of New Material

In order to develop the algebraic equivalent for the AND, the OR, and the EXCLUSIVE OR properties of logic, the following method is utilized. Draw a box, and label it as follows:



THE KRUPP BOX

This box is called the Krupp Box because a student at George Washington High School in Denver, Colorado, Louie Krupp, invented this method for finding the algebraic equivalences to logical expressions.

To find the polynomial which represents the truth table for the connectives, AND, OR, or EXCLUSIVE OR let P and Q represent true and (1-P) and (1-Q) represent false. Now fill in the Krupp Box with the appropriate value of the truth table for the proper connective. To express the polynomial find the sum of the areas of the "TRUE" boxes.

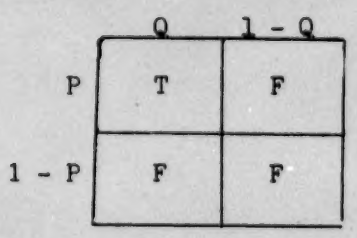
Example 1:

In order to find the algebraic equivalence for the logical connective AND.

THE POLYNOMIAL FOR AND

P	Q	P∧Q
T	T	T
T	F	F
F	T	F
F	F	F

Truth Table for AND (∧)



Krupp Box for AND

AND(P,Q) = PQ (i.e., the area of the true box is P • Q)

in BASIC-DEF FNA(P,Q) = P * Q

© '74 U of DML

THE POLYNOMIAL FOR OR

P	Q	P ∨ Q
T	T	T
T	F	T
F	T	T
F	F	F

Truth Table for OR (∨).

	Q	1 - Q
P	T	T
1 - P	T	F

Krupp Box for OR

OR (P,Q) = PQ + P(1 - Q) + Q(1 - P) = -PQ + P + Q

(The three terms express the areas of the three boxes that have the letter T in them)

in BASIC-DEF FNO (P,Q) = -P * Q + P + Q

THE POLYNOMIAL FOR EXCLUSIVE OR

P	Q	P ⊻ Q
T	T	F
T	F	T
F	T	T
F	F	F

Truth Table for EXCLUSIVE OR(⊻)

	Q	1 - Q
P	F	T
1 - P	T	F

Krupp Box for EXCLUSIVE OR

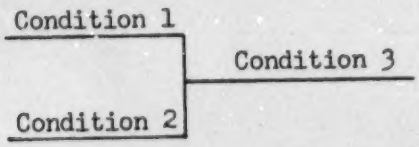
EXCLUSIVE OR (P,Q) = P(1-Q) + (1-P) (Q) = P - PQ + Q - PQ

= (P+Q - 2PQ)

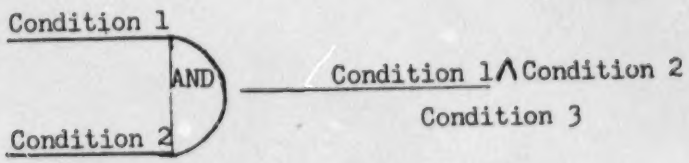
in BASIC-DEF FNE (P,Q) = P + Q - 2*P*Q

© '74 U of DML

3. An application of the above is to simulate simple machinery operations. A gum machine won't vend a package of gum unless the money has been deposited AND a selection has been made. The use of AND requires that two conditions have to be met. Because of the AND requirement the gum will not be vended unless both conditions one and two are met as illustrated below:



If Condition 1 AND Condition 2 are both true, then Condition 3 will be true, if either Condition 1 or Condition 2 is false, then Condition 3 will be false. These conditions represent the fact that Condition 1 has been met by a true or 1 and not met by a false or a 0. A better graphical representation of the above is:

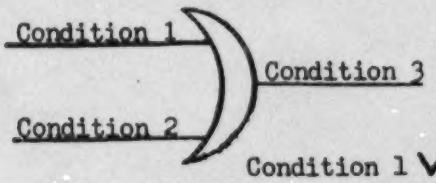


The symbol D is called an AND gate. The conditions enter the flat side, on the left, and the result of the AND of those conditions exits the round side on the right.

In another situation one might want Condition 3 to be on, if either Condition 1 OR 2 were true. For instance, in molding objects, Condition 3, the opening of a mold, may not be continued if the temperature was

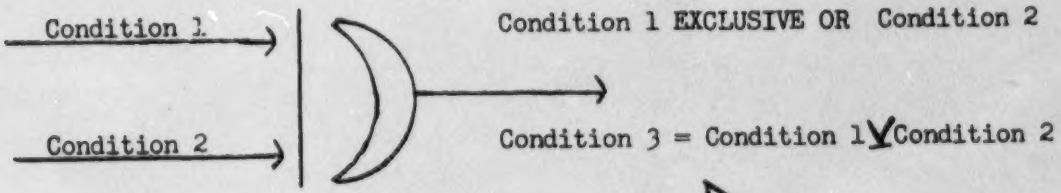
© '74 U of DML

above a predetermined point, OR material is being poured into a mold. The symbol to represent this OR condition appears below:



This symbol is called an OR gate. The conditions enter on the left. The result of the OR of those conditions exits on the right.

Another logical gate which performs the functions illustrated in the truth tables is the EXCLUSIVE OR gate. The gate for EXCLUSIVE OR appears below:



The gate for negation (inverter) is

4. The logical polynomials developed from the Krupp Box can be used in the computer to simulate the logical gates described above.

An example of a Circuit diagram and its translation into a computer model follows. This is an example of a simple safe that has four buttons. The correct choice of buttons, A & C, should open the safe. Any other choice of buttons should sound the alarm.

© '74 U of DML

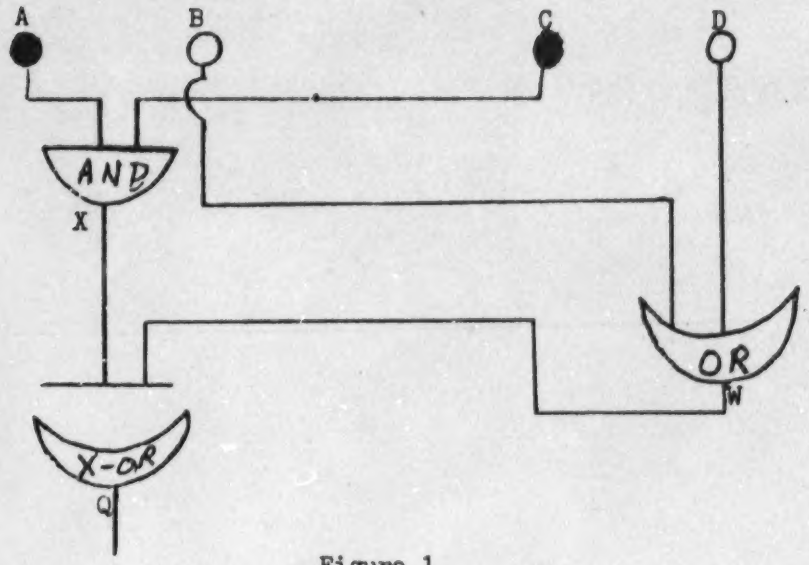


Figure 1

(That is, when Q is false (0), the alarm rings; when Q is true (1), the safe door opens). The computer program that models this circuit needs at point X an AND gate, at point W an OR gate, and at point Q, an EXCLUSIVE OR, and the program that models the diagram above is as follows:

```
10  DEF FNA(P,Q)=P*Q
20  DEF FNE(P,Q)=P+Q-2*P*Q
30  DEF FNO(P,Q)=P+Q-P*Q
40  REM VALUES FOR BUTTONS, ON IS 1 AND OFF IS 0.
50  INPUT A,B,C,D
60  LET X=FNA(A,C)
70  LET W=FNO(B,D)
80  LET Q=FNE(X,W)
90  PRINT " THE VALUES OF A B C AND D ARE "JA;B;C;D;" THEREFORE"
100  IF Q=0 THEN 130
110  PRINT " THE SAFE DOOR OPENS "
120  GO TO 140
130  PRINT " THE ALARM RINGS"
140  END
```

In order to test this program all combinations of A,B,C, and D in the on and off conditions were tested by using four (FOR - NEXT) loops in statements 10 thru 40 (see the program and output on the next page.)

© '74 U of DML

```
4   DEF FNA(P,Q)=P*Q
6   DEF FNO(P,Q)=P+Q-P*Q
8   DEF FNE(P,Q)=P+Q-2*P*Q
10   FOR A=0 TO 1
20   FOR B=0 TO 1
30   FOR C=0 TO 1
40   FOR D=0 TO 1
50   LET X=FNA(A,C)
60   LET W=FNO(B,D)
70   LET Q=FNE(X,W)
90   PRINT " THE VALUE OF A,B,C AND D ARE";A;B;C;D;"THEREFORE"
100   IF Q=0 THEN 130
110   PRINT " THE SAFE DOOR OPENS "
120   GO TO 140
130   PRINT " THE ALARM RINGS"
140   NEXT D
150   NEXT C
160   NEXT B
170   NEXT A
180   END
```

```
 1.  THE VALUE OF A,B,C AND D ARE  0  0  0  0  THEREFORE
     THE ALARM RINGS
 2.  THE VALUE OF A,B,C AND D ARE  0  0  0  1  THEREFORE
     THE SAFE DOOR OPENS
 3.  THE VALUE OF A,B,C AND D ARE  0  0  1  0  THEREFORE
     THE ALARM RINGS
 4.  THE VALUE OF A,B,C AND D ARE  0  0  1  1  THEREFORE
     THE SAFE DOOR OPENS
 5.  THE VALUE OF A,B,C AND D ARE  0  1  0  0  THEREFORE
     THE SAFE DOOR OPENS
 6.  THE VALUE OF A,B,C AND D ARE  0  1  0  1  THEREFORE
     THE SAFE DOOR OPENS
 7.  THE VALUE OF A,B,C AND D ARE  0  1  1  0  THEREFORE
     THE SAFE DOOR OPENS
 8.  THE VALUE OF A,B,C AND D ARE  0  1  1  1  THEREFORE
     THE SAFE DOOR OPENS
 9.  THE VALUE OF A,B,C AND D ARE  1  0  0  0  THEREFORE
     THE ALARM RINGS
10.  THE VALUE OF A,B,C AND D ARE  1  0  0  1  THEREFORE
     THE SAFE DOOR OPENS
11.  THE VALUE OF A,B,C AND D ARE  1  0  1  0  THEREFORE
     THE SAFE DOOR OPENS
12.  THE VALUE OF A,B,C AND D ARE  1  0  1  1  THEREFORE
     THE ALARM RINGS
13.  THE VALUE OF A,B,C AND D ARE  1  1  0  0  THEREFORE
     THE SAFE DOOR OPENS
14.  THE VALUE OF A,B,C AND D ARE  1  1  0  1  THEREFORE
     THE SAFE DOOR OPENS
15.  THE VALUE OF A,B,C AND D ARE  1  1  1  0  THEREFORE
     THE ALARM RINGS
16.  THE VALUE OF A,B,C AND D ARE  1  1  1  1  THEREFORE
     THE ALARM RINGS
```


© '74 U of DML

Notice that, although the safe was supposed to open only for condition 11 (1,0,1,0), it also opened for conditions 2,4,5,6,7,8,9,10,13,and 14. Obviously the first design was poor. The simulation of the model on the computer indicated that the design was faulty. In order to correct this design, a truth table for the buttons and each logical gate was printed out. The design has to be corrected so that the alarm will ring for conditions 2,4,5,6,7,8,10,12,and 14, instead of the safe opening as in the original design.

We use the program below to print out the truth table.

```
1   DEF FNA(P,Q)=P*Q
2   DEF FNE(P,Q)=P+Q-2*P*Q
3   DEF FNO(P,Q)=P+Q-P*Q
5   PRINT " A";" B";" C";" D";" X";" W";" Q"
7   PRINT
10   FOR A=0 TO 1
20   FOR B=0 TO 1
30   FOR C=0 TO 1
40   FOR D=0 TO 1
80   REM VALUES FOR BUTTONS, ON REPRESENTED BY 1, OFF BY 0
90   LET X=FNA(A,C)
100   LET W=FNO(B,D)
110   LET Q=FNE(X,W)
120   PRINT A;B;C;D;X;W;Q
140   NEXT D
150   NEXT C
160   NEXT B
170   NEXT A
180   END
```

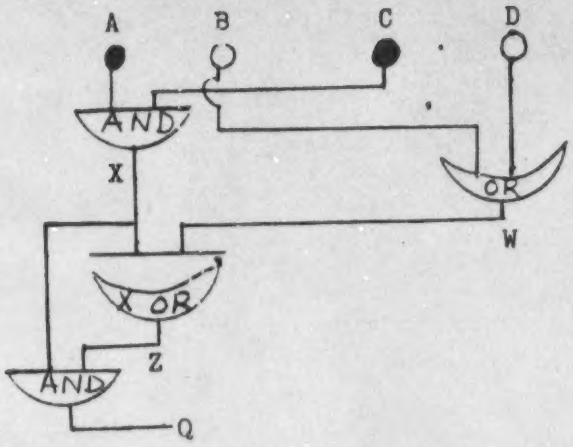
© '74 U of DML

The program produced the following table:

	A	B	C	D	X	W	Q
1.	0	0	0	0	0	0	0
2.	0	0	0	1	0	1	1
3.	0	0	1	0	0	0	0
4.	0	0	1	1	0	1	1
5.	0	1	0	0	0	1	1
6.	0	1	0	1	0	1	1
7.	0	1	1	0	0	1	1
8.	0	1	1	1	0	1	1
9.	1	0	0	0	0	0	0
10.	1	0	0	1	0	1	1
11.	1	0	1	0	1	0	1
12.	1	0	1	1	1	1	0
13.	1	1	0	0	0	1	1
14.	1	1	0	1	0	1	1
15.	1	1	1	0	1	1	0
16.	1	1	1	1	1	1	0

Looking at the above table, we see that the AND gate at X is 1 when the correct combination 1 0 1 0, in row 11, is pushed. If either B or D is pushed when both A and C are pushed, the alarm goes off and the thief is caught. Checking the results if either B or D is pushed, we see that if W (the result of B OR D) is 1 when X (the result of A AND C) is 0, the door opens even though it shouldn't. The problem arises because X and W are fed into an EXCLUSIVE OR gate which does not discriminate whether it is X or W that is on (1). It opens if either of them is 1. Therefore, we must modify our circuit to make this distinction. The following diagram is one way to make the needed modification.

© '74 U of DML



The door opens if Q = 1.

The alarm rings if Q = 0.

The program to test the above diagram is:

```
10   DEF FNA(P,Q)=P*Q
20   DEF FNO(P,Q)=P+Q-P*Q
30   DEF FNE(P,Q)=P+Q-2*P*Q
40   FOR A=0 TO 1
50   FOR B=0 TO 1
60   FOR C=0 TO 1
70   FOR D=0 TO 1
80   LET Q=FNA(FNA(A,C),FNE(FNA(A,C),FNO(B,D)))
90   IF Q=1 THEN 120
100   PRINT " THE ALARM IS RINGING"
110   GO TO 130
120   PRINT " WELCOME, THE DOOR IS OPENING "
130   PRINT " A  B  C  D  ";A;B;C;D
140   NEXT D
150   NEXT C
160   NEXT B
170   NEXT A
180   END
```

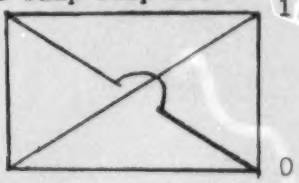
Statements 40-70 and 140-170 merely provide for all possible combinations of the switches being on and off together.

Many circuits require counters, such as the safe that allows the opener several trials before the alarm is set off. In our previous

© '74 U of DML

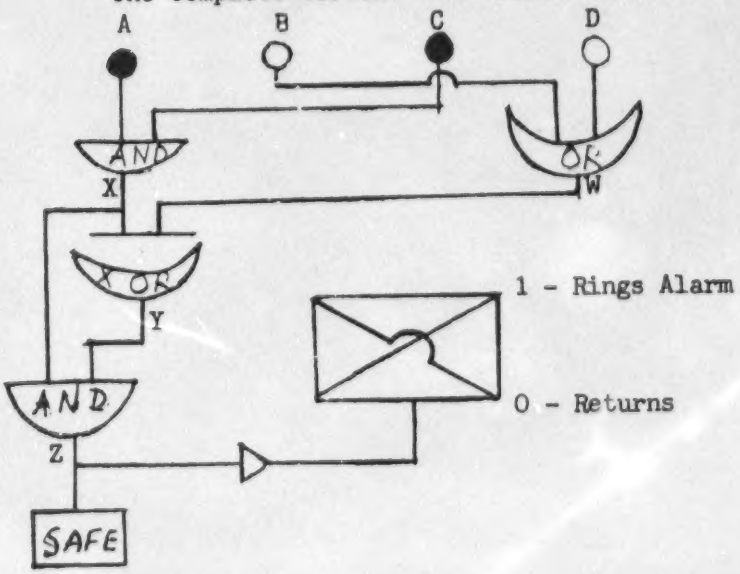
examples let us suppose that two trials are allowed before the alarm is sounded. By use of a device called a flip-flop this can be accomplished.

A symbol for a flip-flop is:



A flip-flop is a device which is in one of two states. We will denote the two states by zero and one. When the flip-flop is in state zero, it changes to state one when an impulse is passed to it, however, in the process, it executes the condition assigned to the zero state. Similarly when the flip-flop is in state one an impulse changes it to state zero and in the process executes the condition assigned to state one. A series of these flip-flops will produce a binary counter, therefore, the choice of this term is analogous with the word counter and is used to control the number of trials in opening the safe before the alarm rings.

The complete circuit is illustrated below:



NOTE:
X = AND (A,C)
W = OR (B,D)
Y = XOR (X,W)
Z = AND (X,Y)

© '74 U of DML

Program for complete circuit is illustrated below:

```
10  LET F=0
20  DEF FNA (P,Q)=P*Q
30  DEF FNO (P,Q)=P+Q-P*Q
40  DEF FNE (P,Q)=P+Q-2*P*Q
50  FOR A = 0 TO 1
60  FOR B = 0 TO 1
70  FOR C = 0 TO 1
80  FOR D = 0 TO 1
90  PRINT " A B C D " A;B;C;D
100  LET Z= FNA(FNA(A,C),FNE(FNA(A,C),FNO(B,D)))
110  LET Q=1-Z
120  IF Z=1 THEN 230
130  IF F=1 THEN 170
140  LET F=1-F
150  GO TO 210
160  IF Q=1 THEN 210
170  PRINT " YOU BLEW IT. THE ALARM IS RINGING"
180  PRINT
190  LET F=0
200  GO TO 260
210  PRINT " PLEASE TRY AGAIN"
220  GO TO 260
230  PRINT " THE DOOR IS OPENING, WELCOME"
240  LET F=0
250  PRINT
260  NEXT D
270  NEXT C
280  NEXT B
290  NEXT A
300  END
```


© '74 U of DML

Printout using flip-flop:

```
A B C D  0     0     0     0
PLEASE TRY AGAIN
A B C D  0     0     0     1
YOU BLEW IT. THE ALARM IS RINGING

A B C D  0     0     1     0
PLEASE TRY AGAIN
A B C D  0     0     1     1
YOU BLEW IT. THE ALARM IS RINGING

A B C D  0     1     0     0
PLEASE TRY AGAIN
A B C D  0     1     0     1
YOU BLEW IT. THE ALARM IS RINGING

A B C D  0     1     1     0
PLEASE TRY AGAIN
A B C D  0     1     1     1
YOU BLEW IT. THE ALARM IS RINGING

A B C D  1     0     0     0
PLEASE TRY AGAIN
A B C D  1     0     0     1
YOU BLEW IT. THE ALARM IS RINGING

A B C D  1     0     1     0
THE DOOR IS OPENING, WELCOME

A B C D  1     0     1     1
PLEASE TRY AGAIN
A B C D  1     1     0     0
YOU BLEW IT. THE ALARM IS RINGING

A B C D  1     1     0     1
PLEASE TRY AGAIN
A B C D  1     1     1     0
YOU BLEW IT. THE ALARM IS RINGING

A B C D  1     1     1     1
PLEASE TRY AGAIN
```

NOTE: In the computer printout above you see that after one incorrect attempt, the opener gets another try. Then after two incorrect attempts the alarm rings. Once the alarm rings the counter is again set to zero (statement 190 in the program). The counter is also reset each time the door is opened (statement 240).

© '74 U of DML

5. One can see that there is another approach for modeling this circuit diagram and that is to express the entire diagram in a circuit polynomial For instance: Z is the AND of X and Y. And if you replace X and Y by their respective representations in terms of A,B,C, and D you can eventually get an entire discription of the circuit in terms of A,B,C, and D. (Refer to the diagram on page 13.)

1) $Z = X \wedge Y$
2) $Y = X \veebar W$
3) $W = B \vee D$
4) $X = A \wedge C$

Observe that by combining the above equations and substituting in their respective values of A,B,C, and D, you can obtain an expression for the entire circuit. The circuit (Z) can be redefined in terms of algebraic representation by using the Krupp Box formulas. (Refer to Suggestion 2, page 5.)

(a) $P \veebar Q = P + Q - 2PQ$
(b) $P \vee Q = P + Q - PQ$
(c) $P \wedge Q = PQ$

By substituting into equation (2) the values for X and W we obtain,

$$Y = (A \wedge C) \veebar (B \vee D),$$

and transforming this equation using the proper Krupp Box formula (EXCLUSIVE OR) yields,

$$Y = (A \wedge C) + (B \vee D) - 2[(A \wedge C)(B \vee D)].$$

However in equation (1) above, $Z = X \wedge Y$ therefore,

$$Z = (A \wedge C) \wedge [(A \wedge C) + (B \vee D) - 2[(A \wedge C)(B \vee D)].$$

If you substitute $\cdot$ for $\wedge$ and + for $\vee$ the polynomial can be evaluated algebraically, thus:

© '74 U of DML

$$Z = A \cdot C \cdot [A \cdot C + (B + D - BD) - 2 \cdot (A \cdot C)(B + D - BD)]$$

$$Z = AC\,[AC + B + D - BD - 2\,(ABC + ACD - ABCD)]$$

$$Z = AC[AC + B + D - BD - 2ABC - 2ACD + 2ABCD)]$$

$$Z = A^2C^2 + ABC + ACD - ABCD - 2A^2BC^2 - 2A^2C^2D + 2A^2BC^2D$$

This equation must be reduced by applying Boolean Algebra

$$Z = AC + ABC + ACD - ABCD - 2ABC - 2ACD + 2ABCD$$

$$Z = AC - ABC - ACD + ABCD$$

$$Z = X \wedge Y = AC - ABC - ACD + ABCD$$

This circuit polynomial can be evaluated for all combinations of true and false (off and on) for A,B,C, and D. That is 2^4 possibilities will provide the 16 possible combinations.

A program to do this follows:

```
FOR A = 0 to 1
FOR B = 0 to 1
FOR C = 0 to 1
FOR D = 0 to 1
PRINT A*C - A*B*C - A*C*D + *A*B*C*D
NEXT D
NEXT C
NEXT B
NEXT A
END
```

This program shows that the entire circuit can be reduced to one polynomial representing the circuit and a printout of the results of the above program follows.

© '74 U of DML

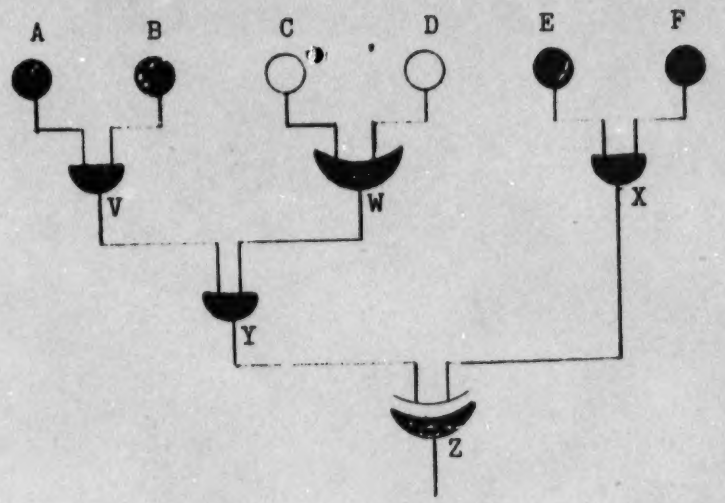
```
5  PRINT " A";" B";" C";" D";" Q"
10  FOR A=0 TO 1
20  FOR B=0 TO 1
30  FOR C=0 TO 1
40  FOR D=0 TO 1
50  LET Q=A*C-A*B*C-A*C*D+A*B*C*D
60  PRINT A;B;C;D;Q
70  NEXT D
80  NEXT C
90  NEXT B
100  NEXT A
110  END
```

A	B	C	D	Q
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

© '74 U of DML

SOLUTIONS TO STUDENT ASSIGNMENT:

1. (a)



1) $Z = Y \veebar X$
2) $Y = V \wedge W$
3) $X = E \wedge F$
4) $W = C \vee D$
5) $V = A \wedge B$

Substitute 4) and 5) into 2)
2) $Y = (A \wedge B) \wedge (C \vee D)$

Substitute 2) and 3) into 1)
1) $Z = ((A \wedge B) \wedge (C \vee D)) \veebar (E \wedge F)$

Use the Krupp Box formulas to complete the polynomial.
Therefore: $Z = (AB)(C+D-CD) + (EF) - 2(EF)[(AB)(C+D-CD)]$
$= (AB-2EF)(C+D-CD) + EF$

© '74 U of DML

(b)

```
10  FOR A=0 TO 1
20  FOR B=0 TO 1
30  FOR C=0 TO 1
40  FOR D=0 TO 1
50  FOR E=0 TO 1
60  FOR F=0 TO 1
70  LET S=(A*B-2*E*F)*(C+D-C*D)+E*F
80  IF S=0 THEN 110
90  PRINT A;B;C;D;E;F;" WILL COMPLETE THE CIRCUIT"
100  GO TO 120
110  PRINT A;B;C;D;E;F;" WILL NOT COMPLETE THE CIRCUIT"
120  NEXT F
130  NEXT E
140  NEXT D
150  NEXT C
160  NEXT B
170  NEXT A
180  END
```

```
 0  0  0  0  0  0   WILL NOT COMPLETE THE CIRCUIT
 0  0  0  0  0  1   WILL NOT COMPLETE THE CIRCUIT
 0  0  0  0  1  0   WILL NOT COMPLETE THE CIRCUIT
 0  0  0  0  1  1   WILL COMPLETE THE CIRCUIT
 0  0  0  1  0  0   WILL NOT COMPLETE THE CIRCUIT
 0  0  0  1  0  1   WILL NOT COMPLETE THE CIRCUIT
 0  0  0  1  1  0   WILL NOT COMPLETE THE CIRCUIT
 0  0  0  1  1  1   WILL COMPLETE THE CIRCUIT
 0  0  1  0  0  0   WILL NOT COMPLETE THE CIRCUIT
 0  0  1  0  0  1   WILL NOT COMPLETE THE CIRCUIT
 0  0  1  0  1  0   WILL NOT COMPLETE THE CIRCUIT
 0  0  1  0  1  1   WILL COMPLETE THE CIRCUIT
 0  0  1  1  0  0   WILL NOT COMPLETE THE CIRCUIT
 0  0  1  1  0  1   WILL NOT COMPLETE THE CIRCUIT
 0  0  1  1  1  0   WILL NOT COMPLETE THE CIRCUIT
 0  0  1  1  1  1   WILL COMPLETE THE CIRCUIT
 0  1  0  0  0  0   WILL NOT COMPLETE THE CIRCUIT
 0  1  0  0  0  1   WILL NOT COMPLETE THE CIRCUIT
 0  1  0  0  1  0   WILL NOT COMPLETE THE CIRCUIT
 0  1  0  0  1  1   WILL COMPLETE THE CIRCUIT
 0  1  0  1  0  0   WILL NOT COMPLETE THE CIRCUIT
 0  1  0  1  0  1   WILL NOT COMPLETE THE CIRCUIT
 0  1  0  1  1  0   WILL NOT COMPLETE THE CIRCUIT
 0  1  0  1  1  1   WILL COMPLETE THE CIRCUIT
 0  1  1  0  0  0   WILL NOT COMPLETE THE CIRCUIT
 0  1  1  0  0  1   WILL NOT COMPLETE THE CIRCUIT
 0  1  1  0  1  0   WILL NOT COMPLETE THE CIRCUIT
 0  1  1  0  1  1   WILL COMPLETE THE CIRCUIT
```


© '74 U of DML

(b) Cont'd

0	1	1	1	0	0	WILL NOT COMPLETE THE CIRCUIT
0	1	1	1	0	1	WILL NOT COMPLETE THE CIRCUIT
0	1	1	1	1	0	WILL NOT COMPLETE THE CIRCUIT
0	1	1	1	1	1	WILL COMPLETE THE CIRCUIT
1	0	0	0	0	0	WILL NOT COMPLETE THE CIRCUIT
1	0	0	0	0	1	WILL NOT COMPLETE THE CIRCUIT
1	0	0	0	1	0	WILL NOT COMPLETE THE CIRCUIT
1	0	0	0	1	1	WILL COMPLETE THE CIRCUIT
1	0	0	1	0	0	WILL NOT COMPLETE THE CIRCUIT
1	0	0	1	0	1	WILL NOT COMPLETE THE CIRCUIT
1	0	0	1	1	0	WILL NOT COMPLETE THE CIRCUIT
1	0	0	1	1	1	WILL COMPLETE THE CIRCUIT
1	0	1	0	0	0	WILL NOT COMPLETE THE CIRCUIT
1	0	1	0	0	1	WILL NOT COMPLETE THE CIRCUIT
1	0	1	0	1	0	WILL NOT COMPLETE THE CIRCUIT
1	0	1	0	1	1	WILL COMPLETE THE CIRCUIT
1	0	1	1	0	0	WILL NOT COMPLETE THE CIRCUIT
1	0	1	1	0	1	WILL NOT COMPLETE THE CIRCUIT
1	0	1	1	1	0	WILL NOT COMPLETE THE CIRCUIT
1	0	1	1	1	1	WILL COMPLETE THE CIRCUIT
1	1	0	0	0	0	WILL NOT COMPLETE THE CIRCUIT
1	1	0	0	0	1	WILL NOT COMPLETE THE CIRCUIT
1	1	0	0	1	0	WILL NOT COMPLETE THE CIRCUIT
1	1	0	0	1	1	WILL COMPLETE THE CIRCUIT
1	1	0	1	0	0	WILL COMPLETE THE CIRCUIT
1	1	0	1	0	1	WILL COMPLETE THE CIRCUIT
1	1	0	1	1	0	WILL COMPLETE THE CIRCUIT
1	1	0	1	1	1	WILL NOT COMPLETE THE CIRCUIT
1	1	1	0	0	0	WILL COMPLETE THE CIRCUIT
1	1	1	0	0	1	WILL COMPLETE THE CIRCUIT
1	1	1	0	1	0	WILL COMPLETE THE CIRCUIT
1	1	1	0	1	1	WILL NOT COMPLETE THE CIRCUIT
1	1	1	1	0	0	WILL COMPLETE THE CIRCUIT
1	1	1	1	0	1	WILL COMPLETE THE CIRCUIT
1	1	1	1	1	0	WILL COMPLETE THE CIRCUIT
1	1	1	1	1	1	WILL NOT COMPLETE THE CIRCUIT

© '74 U of DML

2. (a) $C = 2 * A * B - A - B + 1$

(b) Equivalence or an inverted EXCLUSIVE OR.

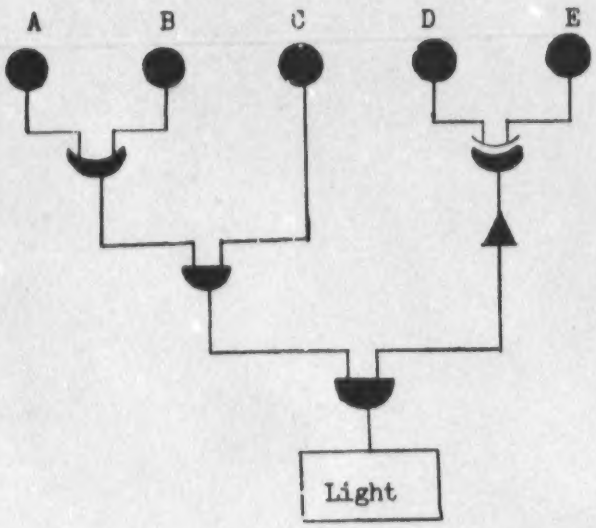
(c)

```
FILE:SPDT - TYPE:BASIC  --06/01/71  1:38 PM.

10  PRINT " IS THE FIRST SWITCH UP OR DOWN"
20  INPUT A$
30  IF A$="UP" THEN 60
40  LET A=0
50  GO TO 7 0
60  LET A=1
70  PRINT " IS THE SECOND SWITCH UP OR DOWN"
80  INPUT B$
90  IF B$="UP" THEN 120
100  LET B=0
110  GO TO 130
120  LET B=1
130  LET C=2*A*B-A-B+1
140  IF C=1 THEN 170
150  PRINT " THE LIGHT IS STILL OFF"
160  GO TO 180
170  PRINT " THE LIGHT IS ON ***********"
180  END

 IS THE FIRST SWITCH UP OR DOWN
?DOWN
 IS THE SECOND SWITCH UP OR DOWN
?DOWN
 THE LIGHT IS ON ***********

 IS THE FIRST SWITCH UP OR DOWN
?DOWN
 IS THE SECOND SWITCH UP OR DOWN
?UP
 THE LIGHT IS STILL OFF

 IS THE FIRST SWITCH UP OR DOWN
?UP
 IS THE SECOND SWITCH UP OR DOWN
?DOWN
 THE LIGHT IS STILL OFF

 IS THE FIRST SWITCH UP OR DOWN
?Y-UP
 IS THE SECOND SWITCH UP OR DOWN
?UP
 THE LIGHT IS ON ***********
```


© '74 U of DML

a) S = [(A + B - A * B) * C] * (2 * D * E - D - E + 1)

b)



c)

```
10  FOR A=0 TO 1 STEP 1
20  FOR B=0 TO 1 STEP 1
30  FOR C=0 TO 1 STEP 1
40  FOR D=0 TO 1 STEP 1
50  FOR E=0 TO 1 STEP 1
60  LET S=((A+B-A*B)B(2*D-D-F+1)
70  IF S=1 THEN 100
80  PRINT A;B;C;D;E;"WILL NOT COMPLETE THE CIRCUIT"
90  GO TO 110
100  PRINT A;B;C;D;E;"WILL COMPLETE THE CIRCUIT"
110  NEXT F
120  NEXT D
130  NEXT C
140  NEXT B
150  NEXT A
160  END
```


© '74 U of DML

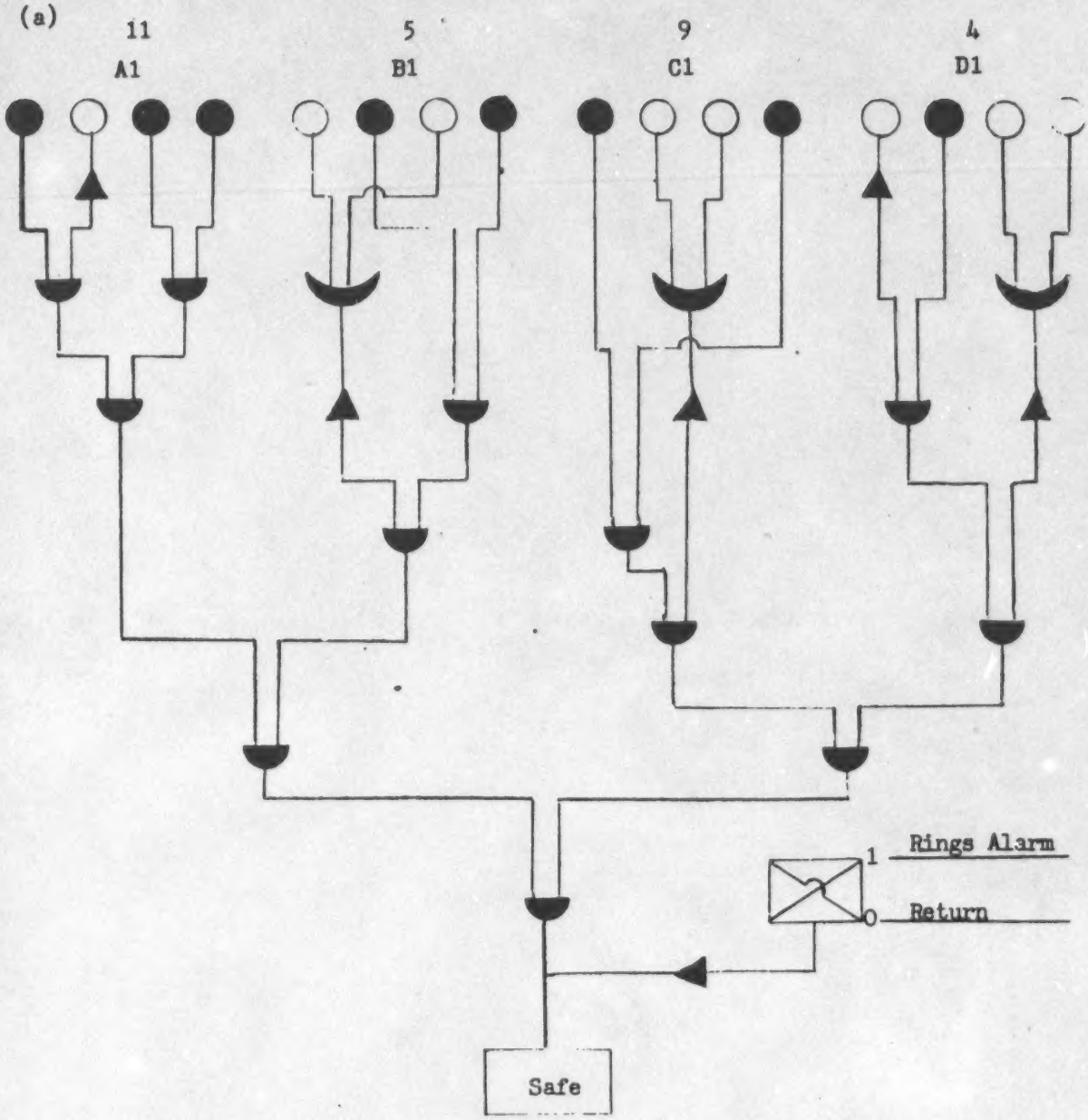
3 (c) Cont'd

0	0	0	0	0	WILL NOT COMPLETE THE CIRCUIT
0	0	0	0	1	WILL NOT COMPLETE THE CIRCUIT
0	0	0	1	0	WILL NOT COMPLETE THE CIRCUIT
0	0	0	1	1	WILL NOT COMPLETE THE CIRCUIT
0	0	1	0	0	WILL NOT COMPLETE THE CIRCUIT
0	0	1	0	1	WILL NOT COMPLETE THE CIRCUIT
0	0	1	1	0	WILL NOT COMPLETE THE CIRCUIT
0	0	1	1	1	WILL NOT COMPLETE THE CIRCUIT
0	1	0	0	0	WILL NOT COMPLETE THE CIRCUIT
0	1	0	0	1	WILL NOT COMPLETE THE CIRCUIT
0	1	0	1	0	WILL NOT COMPLETE THE CIRCUIT
0	1	0	1	1	WILL NOT COMPLETE THE CIRCUIT
0	1	1	0	0	WILL COMPLETE THE CIRCUIT
0	1	1	0	1	WILL NOT COMPLETE THE CIRCUIT
0	1	1	1	0	WILL NOT COMPLETE THE CIRCUIT
0	1	1	1	1	WILL COMPLETE THE CIRCUIT
1	0	0	0	0	WILL NOT COMPLETE THE CIRCUIT
1	0	0	0	1	WILL NOT COMPLETE THE CIRCUIT
1	0	0	1	0	WILL NOT COMPLETE THE CIRCUIT
1	0	0	1	1	WILL NOT COMPLETE THE CIRCUIT
1	0	1	0	0	WILL COMPLETE THE CIRCUIT
1	0	1	0	1	WILL NOT COMPLETE THE CIRCUIT
1	0	1	1	0	WILL NOT COMPLETE THE CIRCUIT
1	0	1	1	1	WILL COMPLETE THE CIRCUIT
1	1	0	0	0	WILL NOT COMPLETE THE CIRCUIT
1	1	0	0	1	WILL NOT COMPLETE THE CIRCUIT
1	1	0	1	0	WILL NOT COMPLETE THE CIRCUIT
1	1	0	1	1	WILL NOT COMPLETE THE CIRCUIT
1	1	1	0	0	WILL COMPLETE THE CIRCUIT
1	1	1	0	1	WILL NOT COMPLETE THE CIRCUIT
1	1	1	1	0	WILL NOT COMPLETE THE CIRCUIT
1	1	1	1	1	WILL COMPLETE THE CIRCUIT

END CIRCUIT 1.3 SEC.

© '74 U of DML

4. (a)



Note: This circuit contains a flip-flop to give the opener a second chance. However, the program given in answer to problem 4(c) allows the opener to continually try new combinations.

(b)
```
A1 = A * (1 - B) * C * D
B1 = B * D * (1 - (A + C - A * C))
C1 = (A * D) * (1 - (B + C - C * B))
D1 = ((1 - A) * B) * (1 - (C + D - D * C))
```


© '74 U of DML

(c)

```
FILE:SAFE - TYPE:BASIC  --06/01/71  2:14 PM.

 10  PRINT " THIS IS AN AUTOMATIC SAFE. INPUT FOUR NUMBERS TO OPEN IT"
 20  PRINT "  IF YOU ARE A BUGULAR HERE IS A HINT "
 30  PRINT "   ( EACH OF THE FOUR NUMBERS IS BETWEEN 0 AND 15 ) "
 40  PRINT " TYPE IN ONE NUMBER PER QUESTION MARK "
 50  FOR Z=1 TO 4
 55  INPUT A(Z)
 60  GO SUB 180
 65  LET A=B(3)
 70  LET B=B(2)
 75  LET C=B(1)
 80  LET D=B(0)
 90  IF Z=4 THEN 160
 91  IF Z=3 THEN 140
 92  IF Z=2 THEN 120
100   LET A1=A*(1-B)*C*D
110   GO TO 280
120   LET B1=B*D*(1-(A+C-A*C))
130   GO TO 280
140   LET C1=(A*D)*(1-(B+C-C*B))
150   GO TO 280
160   LET D1=((1-A)*B)*(1-(C+D-D*C))
170   GO TO 280
175   STOP
180   FOR Y=3 TO 0 STEP -1
190   LET B(Y)=A(Z)-2**Y
200   IF B(Y)>=0 THEN 230
210   LET B(Y)=0
220   GO TO 250
230   LET B(Y)=1
240   LET A(Z)=A(Z)-2**Y
250   NEXT Y
270   RETURN
280   NEXT Z
290   LET A2=A1*B1*C1*D1
295   PRINT
296   PRINT
300   IF A2=0 THEN 330
310   PRINT "  THE SAFE IS OPEN  **********************"
320   GO TO 370
330   PRINT "  INCORRECT, THE ALARM IS RINGING AND THE POLICE ARE ON"
340   PRINT "  THEIR WAY.  DO YOU THINK YOU HAVE ENOUGH TIME TO TRY"
345   PRINT "  AGAIN, THERE ARE ONLY 50,625 POSSIBLE COMBINATIONS "
350   INPUT A$
360   IF A$="YES" THEN 40
370   END
```

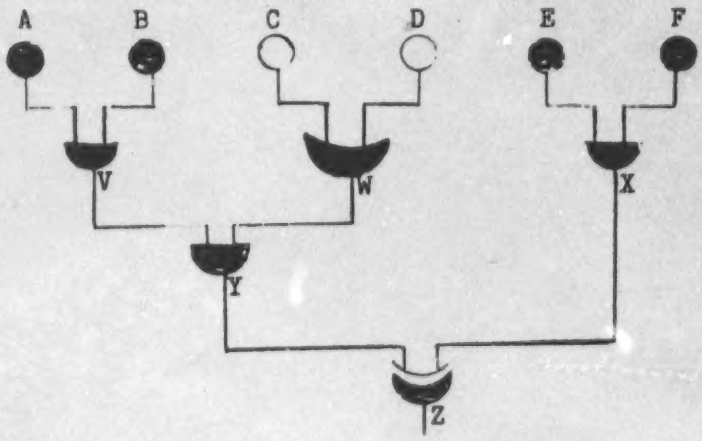

© '74 U of DML

STUDENT ASSIGNMENT

APPLICATION OF LOGIC TO CIRCUIT DESIGN

Purpose: To show students an application of logic and truth tables to the design of circuits. In this endeavor we develop the concept of AND gates, OR gates, and EXCLUSIVE OR gates, then represent the entire circuit by a circuit polynomial.

Problem: 1.

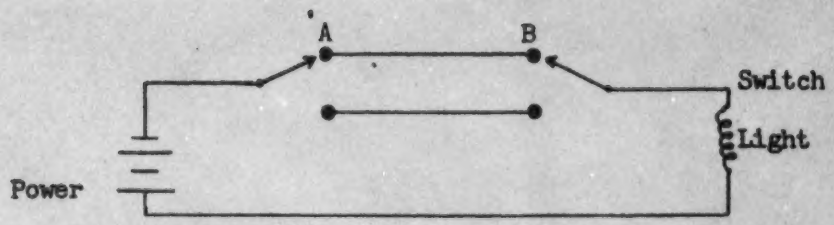


a) Find the circuit polynomial which will represent the above circuit, using the Krupp box formulas.

b) Write a program that will evaluate this circuit for all possible positions of the above switches (A,B,C,D,E, & F).

© '74 U of OML

2.



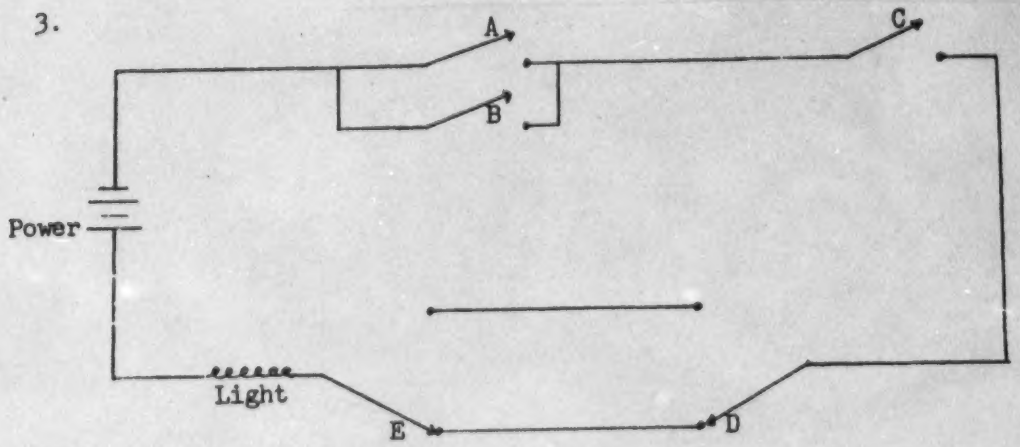
The above diagram represents an electrical circuit containing a three-way switch.

a) Using the Krupp box, develop the circuit polynomial for the above circuit.

b) What logical connective is represented by the Krupp box used above?

c) Write a program using the above polynomial to simulate all possible positions of the switches in this circuit. (The print-out of this program should indicate if the light is on or off.)

3.



a) Using Krupp boxes, develop the circuit polynomial for the above circuit.

© '74 U of DML

b) Transpose the above circuit into a circuit diagram, using the proper symbols for the connectives (AND , OR , EXCLUSIVE OR) presented in this unit.

c) Write a program using the above polynomial to simulate all possible positions of the switches (A,B,C,D,& E) in the above diagram. (The print-out of this program should indicate if the light is on or off.)

Optional

4. Design a safe that will open for the given combination of 11, 5, 9, 4.

a) First construct a circuit diagram as discussed in this unit, using the proper connectives. This diagram must be constructed in four parts, where each section will represent a base ten number converted to its binary representation. Next, connect the individual parts together to the safe to complete the diagram.

b) Represent the diagram in exercise(a) as four separate circuit polynomials, using the Krupp box formulas.

c) Write a program that will accept a combination (of four numbers) in base ten and convert it to base two. Then check the base two combination against the circuit polynomial. Your print-out should show if it is correct or incorrect. If it is incorrect, return to the input statement to repeat this procedure for another combination.

d) How many possible combinations are there?

Hint: Develop a sub-routine that will convert base ten to base two.

© '74 U of DML

I. What is a Boolean Algebra?

Information Card

Logic seems to have some properties common to the algebra of real numbers. For example, if a and b are real numbers, then $a + b = b + a$. Likewise, if p and q are statements, then $p \vee q \leftrightarrow q \vee p$. An investigation of these properties by Englishman George Boole led to the following list of properties.

Let B be a set of elements a, b, c, ... and let $\oplus$ and $\otimes$ be two binary operations defined on B such that for any $a, b, c \in B$:

		$\oplus$	$\otimes$
1.	Closure Laws	$a \oplus b \in B$	$a \otimes b \in B$
2.	Commutative Laws	$a \oplus b = b \oplus a$	$a \otimes b = b \otimes a$
3.	Associative Laws	$(a \oplus b) \oplus c = a \oplus (b \oplus c)$	$(a \otimes b) \otimes c = a \otimes (b \otimes c)$
4.	Distributive Laws	$a \oplus (b \otimes c) = (a \oplus b) \otimes (a \oplus c)$	$a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$
5.	Identity Laws	There exists an identity $0 \in B$ such that $a \oplus 0 = a$	There exists an identity $1 \in B$ such that $a \otimes 1 = 1$
6.	Complementation Laws	For every $a \in B$ there exists $\sim a \in B$ such that $a \oplus \sim a = 1$	For every $a \in B$ there exists $\sim a \in B$ such that $a \otimes \sim a = 0$

Then $(B, \oplus, \otimes)$ is called a Boolean Algebra.

© '74 U of DML

Is $(P, \wedge, \vee)$ a Boolean Algebra?

Exercise

Rewrite the Boolean Algebra Laws for the logic system $(P, \vee, \wedge)$ where P is the set of statements generated by p, q, r,... . Let t be a tautology (statement that's always true) and f be a contradiction (statement that's always false).

© '74 U of DML

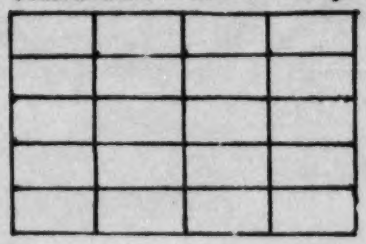
Prove Logic a Boolean Algebra

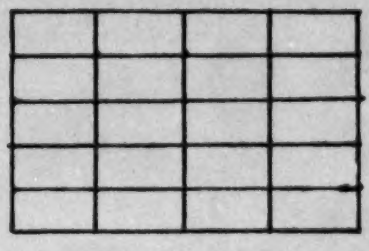
Exercises

Show that the Boolean Algebra Laws for $(P, \vee, \wedge)$ are true.

1. Closure Laws: Let $p, q \in P$. p and q are logical statements. That is they are sentences with truth value (either true or false). Is $p \vee q \in P$? Is $p \vee q$ a sentence with truth value? Is $p \wedge q \in P$?

2. Commutative Laws: Complete the tables.





Is $p \vee q \leftrightarrow q \vee p$?

Is $p \wedge q \leftrightarrow q \wedge p$?

3. Associative Laws: Prove with truth tables.

4. Distributive Laws: Prove with truth tables.

© '74 U of DML

5. Identity Laws: Complete the tables.

Let the contradiction f be the identity for $\vee$

P	f	$P \vee f$
	F	
	F	

$p \vee f \leftrightarrow p$

Let the tautology t be the identy for $\wedge$

P	t	$P \wedge t$
	T	
	T	

$p \wedge t \leftrightarrow p$

6. Complimentation Laws:

P	$\sim P$	$P \vee \sim P$

$p \vee \sim p \leftrightarrow t$

P	$\sim P$	$P \wedge \sim P$

$p \wedge \sim p \leftrightarrow f$

© '74 U of DML

Is (C,+,*) a Boolean Algebra?

Exercises

Determine whether or not (C,+,*) is a Boolean Algebra where C = {1,0} and + and * are defined as follows:

+	1	0
1	1	1
0	1	0

*	1	0
1	1	0
0	0	0

© '74 U of DML

Is $(S, \cup, \cap)$ a Boolean Algebra?

Exercises

IV. Recall the definitions of union and intersection of sets. Determine whether or not $(S, \cup, \cap)$ is a Boolean Algebra.

© '74 U of DML

Is (R,+,X) a Boolean Algebra?

Exercises

V. Determine whether or not the set of real numbers with the operations addition and multiplication is a Boolean Algebra.

© '74 U of DML

II. Using Boolean Algebra to Prove Theorems

Information Card

Like the rules of the algebra of real numbers, the rules of Boolean Algebra may be used to prove theorems.

Example: Are the statements "It is not true that it is not raining." and "It is raining." equivalent?

Let S: It is raining.

Then $\sim(\sim S)$: It is not true that it is not raining.

Prove $S \leftrightarrow \sim(\sim S)$
(remember t is a tautology and f a contradiction)

Statements	Reasons
1. $S \leftrightarrow S \vee f$	1. Identity $\vee$
2. $S \leftrightarrow S \vee (\sim S \wedge \sim(\sim S))$	2. Complimentation $\wedge$
3. $S \leftrightarrow (S \vee \sim S) \wedge (S \vee \sim(\sim S))$	3. Distributive $\vee$ over $\wedge$
4. $S \leftrightarrow t \wedge (S \vee \sim(\sim S))$	4. Complimentation $\vee$
5. $S \leftrightarrow (\sim S \vee \sim(\sim S)) \wedge (S \vee \sim(\sim S))$	5. Complimentation $\vee$
6. $S \leftrightarrow (\sim(\sim S) \vee \sim S) \wedge (\sim(\sim S) \vee S)$	6. Commutative $\vee$
7. $S \leftrightarrow \sim(\sim S) \vee (\sim S \wedge S)$	7. Distributive $\vee$ over $\wedge$
8. $S \leftrightarrow \sim(\sim S) \vee f$	8. Complimentation $\wedge$
9. $S \leftrightarrow \sim(\sim S)$	9. Identity $\vee$

Therefore, you may sleep well tonight knowing assuredly that $S \leftrightarrow \sim(\sim S)$.

© '74 U of DML

Using Boolean Algebra to Prove Theorems
Activity Card

Notice that in Boolean Algebra every statement (axiom or theorem) is paired with another much like it. Each statement has a dual. In any true statement we can exchange $\wedge$ and $\vee$ to make its dual.

Example: $p \wedge (q \vee r) \leftrightarrow (p \wedge q) \vee (p \wedge r)$ Distributive $\wedge$ over $\vee$

$p \vee (q \wedge r) \leftrightarrow (p \vee q) \wedge (p \vee r)$ Distributive $\vee$ over $\wedge$

(dual)

$\sim(p \vee q) \leftrightarrow \sim p \wedge \sim q$ I. DeMorgan's Law

Write dual: II. DeMorgan's Law

You have proven I. DeMorgan's Law by truth tables. Now prove it by Boolean Algebra. We want to show $\sim p \wedge \sim q$ is equivalent to the complement of $p \vee q$. We know 1.) a statement and its complement $\leftrightarrow$ f and 2.) a statement or it's complement $\leftrightarrow$ t.

1.) $(\sim p \wedge \sim q) \wedge (p \vee q)$	$\leftrightarrow (\sim p \wedge \sim q \wedge p) \vee (\sim p \wedge \sim q \wedge q)$	1.
	$\leftrightarrow (\sim p \wedge p \wedge \sim q) \vee (\sim p \wedge \sim q \wedge q)$	2.
	$\leftrightarrow (f \wedge \sim q) \vee (\sim p \wedge f)$	3.
	$\leftrightarrow f \vee f$	4.
	$\leftrightarrow f$	5.

Supply the reason for each step.

2.) $(\sim p \wedge \sim q) \vee (p \vee q)$	$\leftrightarrow (\sim p \vee p \vee q) \wedge (\sim q \vee p \vee q)$	1.
	$\leftrightarrow (\sim p \vee p \vee q) \wedge (\sim q \vee q \vee p)$	2.
	$\leftrightarrow (t \vee q) \wedge (t \vee p)$	3.
	$\leftrightarrow t \wedge t$	4.
	$\leftrightarrow t$	5.

Supply the reason for each step.

The proof of II. DeMorgan's Law does not need to be shown since each statement would be the dual of the above statements and we know these duals to be true statements.

© '74 U of DML

Using Boolean Algebra to Prove Theorems

Exercises

Prove the following theorems.

Theorem 1 $p \vee p \leftrightarrow p$

Statements	Reasons
1. $p \vee f \leftrightarrow p$	1. Identity $\vee$
2.	2.

Theorem 2 $p \wedge p \leftrightarrow p$

Statements	Reasons
1.	1.

© '74 U of DML

Theorem 3 $p \vee t \leftrightarrow t$

Statements	Reasons
1.	1.
2.	2.
3.	3.
4.	4.

Theorem 4 $p \wedge f \leftrightarrow f$

Statements	Reasons
1.	1.

© '74 U of DML

Theorem 5 $p \vee (p \wedge q) \leftrightarrow p$

1. $p \vee (p \wedge q) \leftrightarrow (p \wedge t) \vee (p \wedge q)$	1. Identity $\wedge$

Theorem 6 $p \wedge (p \vee q) \leftrightarrow p$

1. $p \wedge (p \vee q) \leftrightarrow (p \vee f) \wedge (p \vee q)$	1. Identity $\vee$

© '74 U of DML

III. Using Boolean Algebra to Simplify Logical Statements
Information Card

The rules and theorems of Boolean Algebra can be used to simplify Boolean expressions.

Example: The cheerleaders use a cheer which contains the phrase "We win and we fight or we lose." The coach objects to using the term "lose" in a cheer. Can an equivalent phrase be found which doesn't contain the word "lose"?

Let w: We win. Then $\sim w$: We lose.

Let x: We fight.

The original statement is symbollically: $w \wedge (x \vee \sim w)$

$w \wedge (x \vee \sim w) \longleftrightarrow (w \wedge x) \vee (w \wedge \sim w)$	Distributive $\wedge$ over $\vee$
$\longleftrightarrow (w \wedge x) \vee f$	Complimentation $\wedge$
$\longleftrightarrow w \wedge x$	Identity $\vee$
$\longleftrightarrow x \wedge w$	Commutative $\wedge$

The cheerleaders (thanks to the logic class) may use the equivalent phrase "We fight and we win."

© '74 U of DML

Simplifying with Boolean Algebra
Exercises

Simplify each of the following (Remember you can use the Laws of Boolean Algebra, the theorems you proved in the last exercise and De Morgans Laws).

1. $p \vee (p \wedge q)$

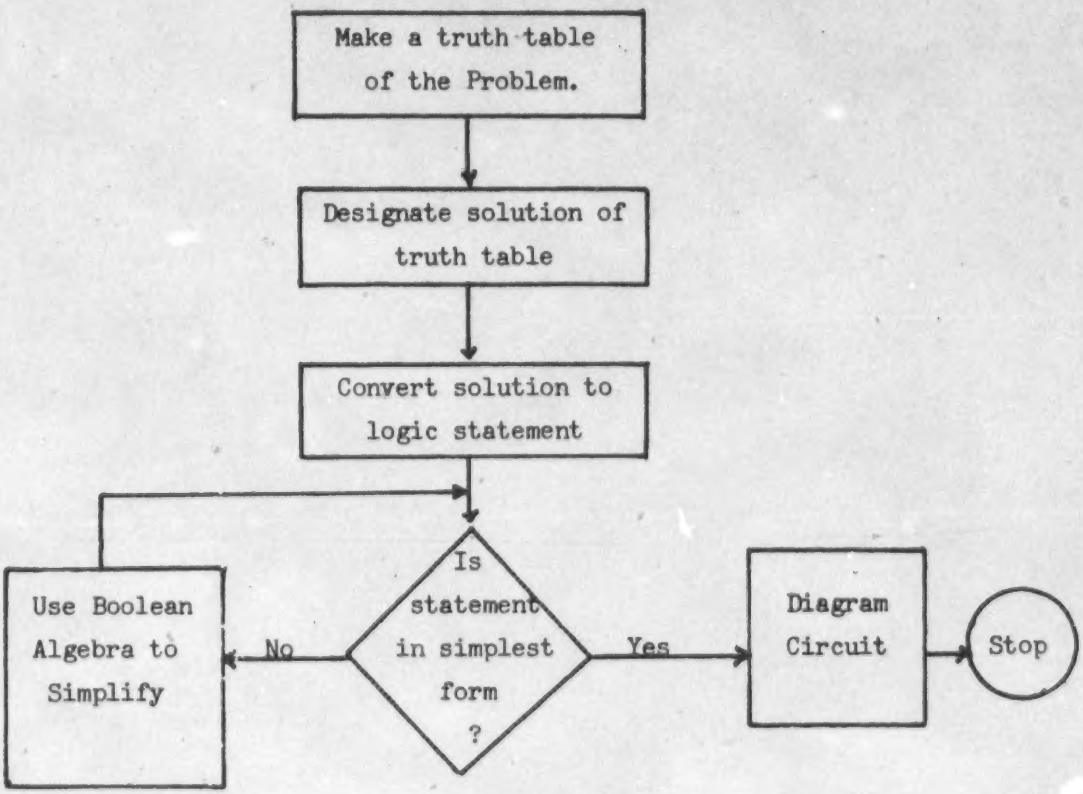
2. $(p \wedge q) \vee [(p \wedge \sim q) \vee (\sim p \wedge \sim q)]$

3. My teacher is boring or my teacher is funny, and my teacher is not boring and my teacher is not funny.

© '74 U of DML

IV. Circuit Problems with Logic and Boolean Algebra
Activity Card

Light switch problems, combination lock problems, and various other similar problems yield to systematic solution. Simply follow this procedure:



© '74 U of DML

Example: In a new house there is a room containing two distinct entraces: A and B. Each entrance is to have a switch to control one central ceiling light. Design a circuit which will allow you to turn the light on or off by the use of the switch to which you happen to be closest.

Solution:

Truth table

A	B	light (solution)
1	1	
1	0	
0	1	
0	0	

Solution of truth table

When both switches are off the light should be off.

When one of the switches are on, the light should be on.

When both switches are on, the light should be off.

A	B	light (solution
1	1	0
1	0	1
0	1	1
0	0	0

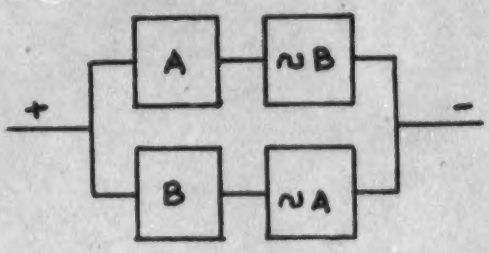
© '74 U of DML

Logic statement of solution (when is light on).

$(A \wedge \sim B) \vee (B \wedge \sim A)$

The statement requires no simplification.

Circuit diagram



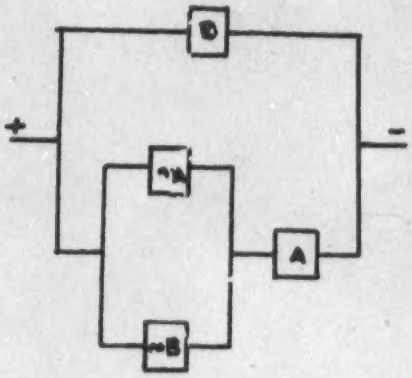
1. Wire this circuit and see if it works.
2. Change the two light switch problem to include three entrances and three light switches. Follow the same procedure to solve the problem.
3. A farmer must transfer a fox, a pig, and a bushel of apples across a lake in a small boat which allows for only one item in addition to himself. The farmer may not leave the pig alone with the apples or the apples will be converted to pork. Likewise, if the fox were left alone with the pig the result would be a fat fox and a very lean pig. You are to help the poor farmer by designing an alarm system which will light a light whenever a potentially dangerous situation will occur.
4. Try a four light switch problem.

© '74 U of DML

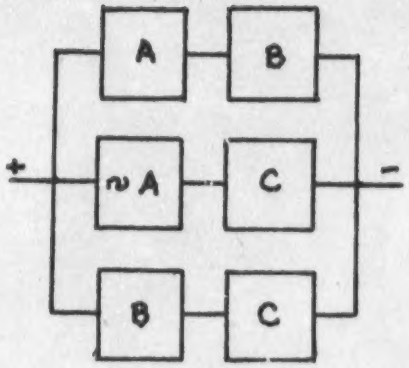
Exercises

For each of the following write a logic statement, simplify using Boolean Algebra, then draw the simpler circuit.

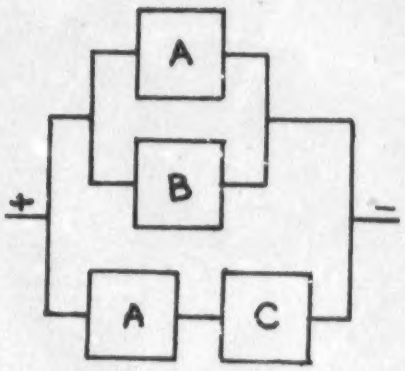
1.



2.

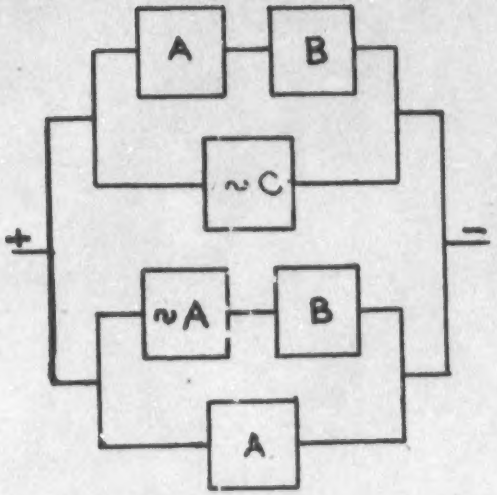


3.

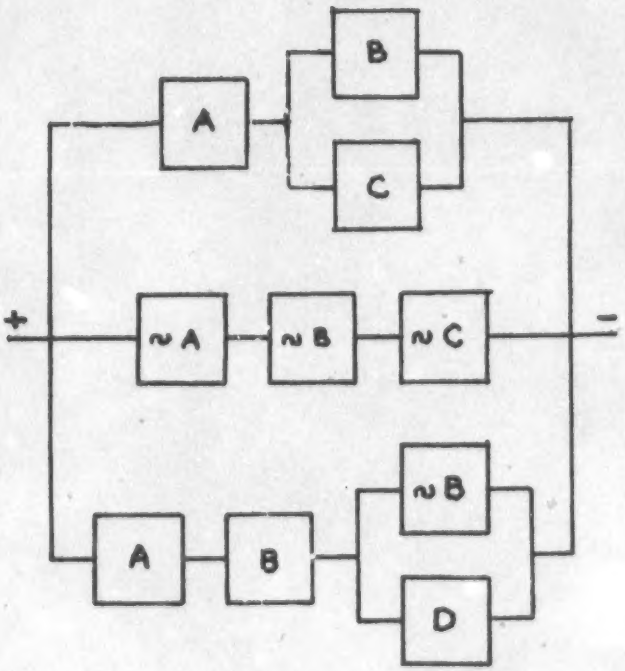


© '74 U of DML

4.

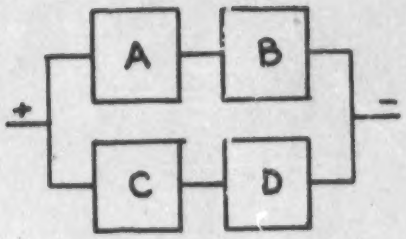


5.

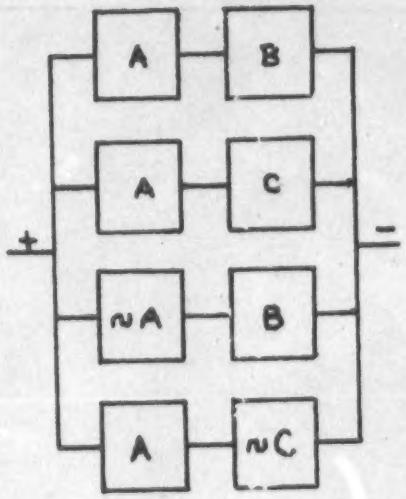


© '74 U of DML

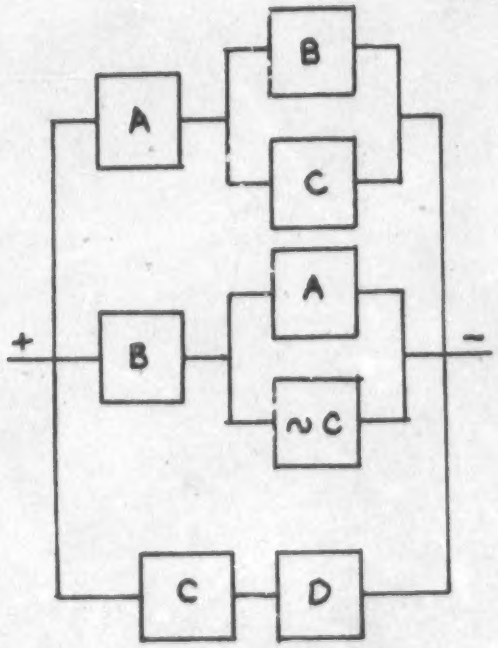
6.



7.

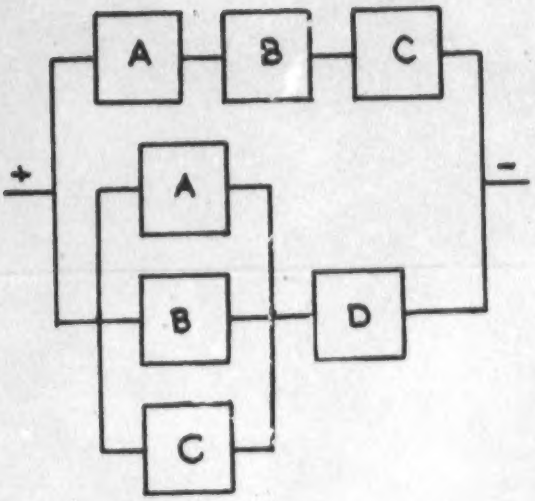


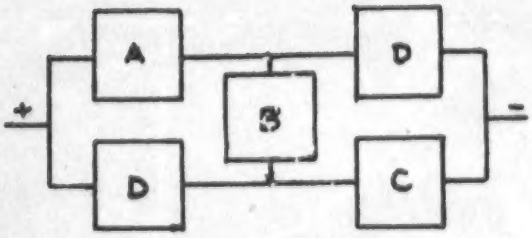
8.



© '74 U of DML

9. Show these two circuits below are equivalent.





© '74 U of DML

Simplify each statement using Boolean Algebra.

10. $(\sim A \wedge \sim B \wedge D) \vee (B \wedge \sim C \wedge \sim D) \vee (A \wedge \sim B \wedge D) \vee (B \wedge C \wedge D)$
11. $(\sim A \wedge \sim B \wedge D) \vee (B \wedge \sim C \wedge \sim D) \vee (A \wedge \sim B \wedge D) \vee (B \wedge C \wedge D) \vee (B \wedge D)$
12. $(A \vee C \vee \sim D) \wedge (A \vee \sim B \vee C)$
13. $(A \vee C \vee \sim D) \wedge (A \vee \sim B \vee C) \wedge (B \vee C \vee D) \wedge (B \vee \sim C \vee D)$

© '74 U of DML

Supplementary Exercises

1. A, B, C are sets. Suppose $A \subset B$ and $B \cap C = \emptyset$. Prove $A \cap C = \emptyset$ by the indirect method.

2. Construct indirect proof for each of the following
   a) If x^2 is odd, then x is odd.
   b) If I am to pass this course, I must do homework regularly.
   c) If he earns a great deal of money (more than $30,000) he is not a college professor.

3. Give a symbolic analysis of the following argument:
   "If he is to succeed, he must be both competent and lucky. Because, if he is not competent, then it is impossible for him to succeed. If he is not lucky, something is sure to go wrong."

4. Give a symbolic analysis of the following argument:
   "If Jones is the murderer, then he knows the exact time of death and the murder weapon. Therefore, if he does not know the exact time or does not know the weapon, then he is not the murderer."

5. Verify that (a), (b), and (c) are equivalent to $p \rightarrow q$.
   a) $(p \wedge \sim q) \rightarrow \sim p$
   b) $(p \wedge \sim q) \rightarrow q$
   c) $(p \wedge \sim q) \rightarrow (r \wedge \sim r)$

6. Supply the conclusion to the following argument, making it a valid argument.
   "If he goes to a party, he does not fail to brush his hair. To look fascinating it is necessary to be tidy. If he is an opium eater, then he has no self-command. If he brushes his hair, he looks fascinating. He wears white kid gloves only if he goes to a party. Having no self-command is sufficient to make one look untidy. Therefore, ..."

7. A logician is captured by a tribe of savages and placed in a jail having two exits. The savage chief offers to the captive the chance to escape: "One of the

© '74 U of DML

doors leads to freedom and the other to certain death. You may leave by either door. To help you in making a decision, two warriors will stay with you and answer one question. I must warn you, however, that one of my warriors always tells the truth, while the other always lies." The chief then leaves. After thinking a moment, the captive asks one question and chooses the door leading to freedom. What did he ask?

The Schaum's Outline Series "Set Theory and Related Topics" by Seymour Lipschutz, pp. 223-224 is an excellent source of supplementary problems.

© '74 U of DML